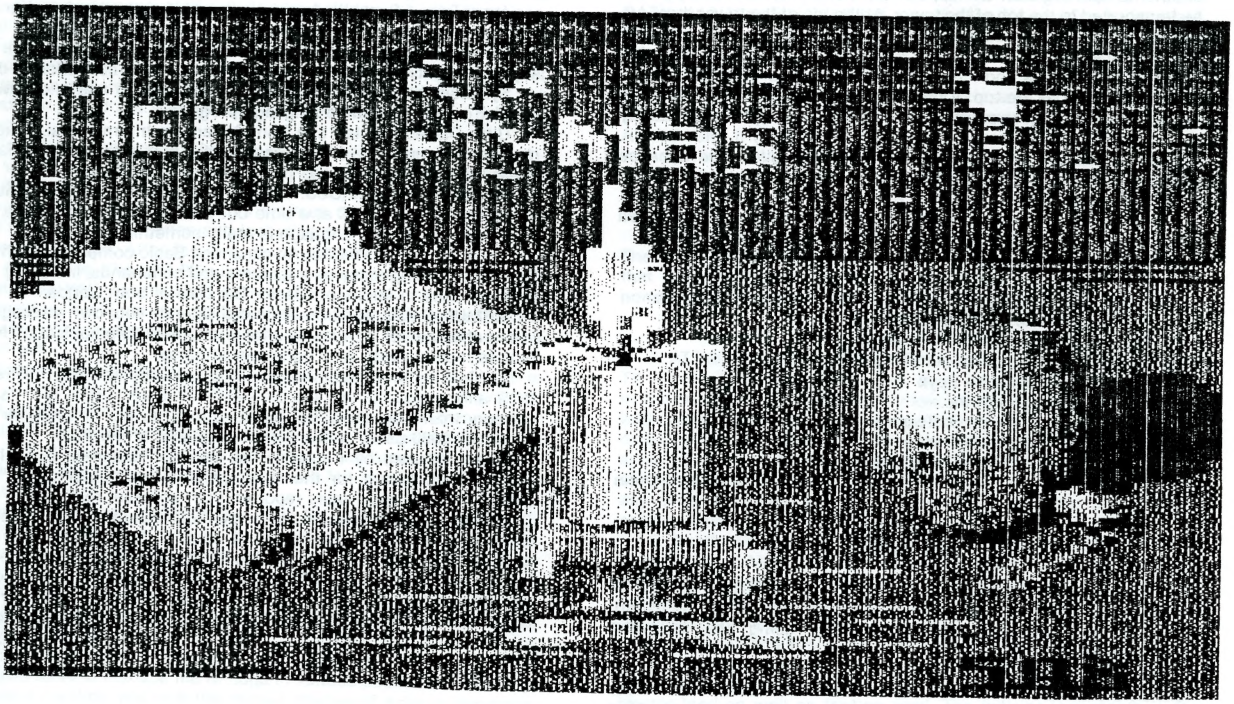
ACE
ATARI COMPUTER ENTHUSIASTS
3662 Vine Maple Dr. Eugene OR 97405

DECEMBER 1983 - JANUARY 1984
Mike Dunn & Jim Bumpas, Editors



-by Scott Berfield

BRYAN'S ARCADE

Quasimodo

This month I recieved a new game called Quasimodo. Quasimodo is a fantastic new game from Synapse. It is written by a fellow teenager named Jon Atack who lives here in Eugene, it will be released soon by Synapse Software.

In the game you are Quasimodo. You have been falsely accused of stealing the king's royal jewels. Of course you didn't do it. Your mission is to rescue the royal jewels from the crooks who took them and bring them back to the king and queen.

There are three levels to Quasimodo.

Level 1

Level 1 is extremely easy. You are automatically transported to the bottom of the tower. Below you are archers climbing up ladders trying to get to the top of the ladders and shooting arrows up at you. You have three piles of rocks you can protect yourself with. You can pick up a rock and throw it down at the archers and they will fall back down the ladders. After you kill all of the archers a ladder leading to the first royal jewel will appear. Climb up and get the jewel and climb back down and go through the teleporting door.

Level 2

On level 2 you have to go through the archers again and then you go back to where the first jewel was and you will see some bells with ropes on them. If you jump on the ropes and move the joystick back and forth little Quasimodo will start swinging on the bells. Once the bell starts ringing you have enough momentum to jump to the next bell. To get the jewel on level 2 you have to go bell by bell and ladder by ladder until you get the jewel. All the time you are swinging, jumping and climbing you have to dodge the bats which constantly try to bite you! After you get the second jewel you have to climb back down to the bottom of the tower and through the teleporting door once again.

Level 3

On level 3 you do exactly the same thing you did on Level 2 until you get to where the second royal jewel was. Once you get there another teleporting door will appear. Go inside this door and you will be transported to the top of the tower. At the top of the tower there are soldiers in little guard booths shooting arrows at you. You are also armed with a bow and an endless supply of arrows so you can also shoot at the soldiers and kill them. There are also guardians of the main royal jewel at the top of the tower. The guardians throw things down at you and if an arrow or a weird thing hits you, you will instantly die.

The sound and graphics are very good like all of the Synapse games. I definitely recommend this game for your friend, kid or any avid video game player.

That's all for the Quasimodo review. Now I will make a list and tell you what I think are good games to get for Christmas!!

Quasimodo by Synapse
Pole Position by Atari
Pinball Construction Set by Bill Budge and Electronic Arts
Arcade Machine by Broderbund Software
Star League Baseball by Gamestar
Jumpman or Jumpman Junior by Epyx
Gateway to Apshai by Epyx
PooYan by Data-Soft Inc.
Blue Max by Synapse Software
Zepplin by Synapse Software

That's all for this month! Happy Holidays!

—Bryan Dunn

ACE

I want to extend season's greetings to ACE members using a 14 line BASIC program. If you like brain teasers, don't look at the listing, and try to write the program yourself using the following specifications.

Using only Atari BASIC, write a program putting small and large text on the screen in GR.0 and border the entire screen with cursor characters. Do not modify the display list and do not use any machine language, but complete the display in less than half of one second.

If you think this is impossible, type in my little BASIC demo. Who says Atari BASIC is slow?

It's been quite a while since I've submitted anything to ACE and I apologize. I've been busy writing for ANTIC, writing The Atari Reference Encyclopedia for The BOOK COMPANY, completing CHATTERBEE for DON'T ASK SOFTWARE, and ROBIN'S HALLOWEEN for PDI. Mike Dunn will be receiving review copies of the book and software in November.

Happy Holidays to you all, —Jerry White

BUMPAS REVIEWS

This is the time of year we at ACE make recommendations of the "10 Best" software titles for the gift-giving season. I've been asked to make the list this year because of all the reviewing I've been doing (I just wrote a book of 200 reviews of Atari software for Consumer Guide).

The 10 best games on my list include:

Excalibur by Chris Crawford for $30 from APX. This game recreates an entire Arthurian environment for the player.

Eastern Front (cartridge) by Chris Crawford for $45 from Atari. World War 2 on your Atari.

M.U.L.E. for $40 from Electronic Arts. For 1-4 players trying to survive on a mining colony in a remote area of the galaxy.

Combat Leader for $40 from Strategic Simulations. Command your own platoon or company in a tactical encounter with the Atari.

Star League Baseball for $32 from Gamestar. Miss your baseball in the winter-time? This has the feel of the real thing, and is an excellent game to boot!

Blue Max for $40 from Synapse. Fly your WWI biplane through enemy defenses to bomb a city. Almost a flight simulator.

Chess for $70 from Odesta. The easiest to use, most flexible and most challenging Chess game for a home computer I've seen.

Murder on the Zinderneuf for $40 from Electronic Arts. A joystick/graphic detective game giving you 40 minutes to solve one of 128 crimes of murder.

Pole Position for $45 from Atari. The popular arcade formula driving game-simulator brought home.

Lode Runner $35 from Broderbund. The latest incarnation of the running, jumping, climbing, digging game with over 150 screens, including user-defined ones.

For non-game software my list of ten includes:

AtariWriter for $99 from Atari. The best word processor for the Atari which is as good as some of the best professional word processors in business use.

FileManager 800+ for $99 from Synapse. A full-functioned database manager which is as easy to use as it is powerful.

JonesTerm 3.6, public domain from your user group. Let your Atari talk to other computers over the phone lines with a modem.

SCOPY, $15 from ACE. You will use this sector copier more than any other utility as you back up and manage your collection of disks.

Synassembler, $49 from Synapse. An easy to use and powerful tool for creating machine language programs.

Ultra Dissassembler, $49 from Adventure International. Take any machine language file and disassemble it with standard Atari labels. Great for studying advanced programming techniques.

Action!, $99 from OSS. A new language on a bank switched cartridge using only 8K at a time but has 24K on the cartridge. This superfast, structured language is really something.

Basic XL, $99 from OSS. An expanded Basic compatible with Atari Basic, but adding many new features and commands, including most features used in Microsoft Basic. The best of both worlds.

ARMUDIC BBS, $99 from Armudic. Operate your own message and information center, or just use it for unattended communication from a remote location with your computer at home.

MYDOS by Wordmark Systems and available from SWP. This new DOS allows you to use double-sided, 80 track disks. It uses less memory than Atari DOS, but works with 8" drives or whatever drive your ATR 8000 uses, or any disk drive made for the Atari.

Note: Zaxxon, by Datasoft, has received Eastman Publishing's Golden Disk Award for the "best personal computer game." This award is based upon sales reported from software retailers.

Broderbund has discovered some "incompatibility during play of 'Operation Whirlwind' with certain boards. The problem has been corrected." I had been experiencing a lock-up during play and thought it was my hardware problem. I'm sure Broderbund will make the correction for you if you use this program and have a Mosaic or other board which is incompatible with the earlier version of their game.

— Jim Bumpas
co-editor

News and Reviews

Mike Dunn, Co-Editor

It is the Holiday season again, so we at ACE wish you all good luck and a happy new year. Because of increasing expenses, we need to increase our dues to $12 a year starting Jan 1 ($20 a year overseas Air. If your renewal is postmarked before, send the current rate. When the postal rate increase takes effect, we will have to boost our subscription rates again.

The new version of Frank Hubband's ARMUDIC BBS will support 1200 baud transmission. It will also allow the use of 64K so we can have many more messages. The bulletin board is in constant use, and some consideration of using a password system for members and limiting access to others has been suggested -- see Larry Gold's article. Larry, besides being SYSOP and Vice-Pres of ACE, is now helping out as the "Lay-Out Editor."

Remember this is the Dec/Jan issue, so the next issue will be in FEB.

New in the Program Library

Our new "Best of ACE #7," covering programs from Oct ot Dec is now ready, as, for an Xmas special, this and all "Best of . . ." series are available for $10 each or $15 for a double sided disk, rather then $15/$20, for orders postmarked before Jan 15. Also new is a new *PILOT #2 Disk* ($10), which includes a very elaborate Geometry program much too long to be in the newsletter and includes extensive directions on the file. This great program was sent to us by Abdulhadi Yassin of San Francisco in appreciation for all the public domain PILOT programs of Ruth Ellsworth. He apparently learned to program by using her programs as tutorials. Thank you Abdulhadi and keep them coming.

In this issue is a game called "Bugs"published both by us and by **Page 6**, a UK Atari magazine (18, Underwood Close, Parkside, Stafford) for the Chirstmas issue. It is a very fine game, but difficult. From our very own, prolific Stan Ockers is "Atrain," a wonderful little educational game for the beginning speller — I love it!! It is really cute.

Also starting in this issue is a new series on the people behind ACE. Because he suggested it, the first autobiography is by Ron Ness, our program exchange librarian. Ron and his son Aaron have been with ACE almost from the very first; they have always been available to help out on everything, like collating the newsletter, licking stamps, or whatever needs to be done. Without members like the Ness', the newsletter would not ever get out — thanks for everything, Ron and Aaron.

For news, I suppose you have all heard TI has thrown in the towel -- this should be a big advantage for the Atari computer. I also hear the new 800XL is very nice — I have not seen one yet, but all I hear about it is good. I also hear from various sources that the more advanced Atari's, like the 1400 series and including the 16 bit one, have been scrapped for sure. After the loss of all that money, Atari will play it safe for a while. The IBM PCjr is now out and it appears to be a high priced dud. With a "chicklet" keyboard and an OS that is specially designed so it will not run much of the outstanding business IBM PC software, it is very expensive for what you get. The Atari seems to be a better bet.

A new magazine for the Atari home computer and the Atari VCS is now out and available. **Hi-Res** is $20 a year for a charter subscription (933 Lee Rd., Suite 325, Orlando, FL 32810. The first issue features that wonderful Doonesbury script advertaizing the Atari peace package. I have to confess I have been a Doonsberry fan since the strip was started and was depressed for a week when Mr. Trudeau went on his 2 year leave.

MovieMaker (Reston, 11480 Sunset Hills Rd., Reston, VA 22090, $60)

This fantastic new program has been a big hit with my family, especially those who have no interest in computers. It allows you to make your own computer generated animated cartoons and movies and it's so easy anyone can do it. An 87 page booklet takes you step by step through the process. There are files of preprogrammed shapes to get you started; but you can easily make your own. To begin your cartoon, you go to the compose utility, and pick your shape. As an example, there is a shape file of dogs in various positions of running. With your joystick, you pick the various shapes of dogs in sequence, then preview the action whenever you want. You can program the sequences with as many as 16 shapes, change colors, speed, etc. When you have your shapes and sequences as you want them, you go to the Record utility, and decide the frame rate, number of frames per shape, jumps per joystick, and loops of each sequence. You then use your joystick to program the action around the screen and record the whole process. When you are finished and all is the way you like, you can smooth the whole thing so it looks like a real motion picture!! And all of the above only takes a few minutes to do!

After you have made a few simple movies, you can start to become a real director. Using the compose utility, you can draw your own shapes and save them to a file. Basic drawing commands allow you two levels of zoom magnification for detailed work. For ease of use, there are duplicating commands and mirror image commands, although we couldn't get the mirror command to work. You can easily fill with colors of your choice. There is even a command to make your joystick a left-handed one for left-handers! There is also a utility for making various backgrounds easily.

Other features include the ability to edit your "finished" movie, go to a specific frame and rewind, mark a frame to stop, freezing a sequence, zooming and expanding shapes to stimulate an object going towards or away from you. Sounds can be added as well as titles. The MovieMaker allows 4 colors on a screen at a time, 9 sequences at one time, 16 frames in a sequence, 64 shapes on a shape page, up to 1/4 of a screen size for an outline, 6 actors and 300 frames per animation.

After you have made your movie, you can give it to friends on an autoboot playback disk. You cannot add it to your own programs as it stands now, but I bet one of our members will figure out a way and write an artilce for ACE. A truly outstanding product, highly recommended for anyone interested in the art of animation or learning it.

Other review copies sent to us for review by Reston this month did not arrive in time for a full review — they will be in the next issue, but here is a short report:

Space Knights by David Heller and Robert Kurcina. David Heller is the author or "Free Software for the Atari," the book that has helped us get many new members and almost overloaded our bulletin board. It is a 156 page science fiction action novel with a number of interesting games on disk involving the reader in the action of the main characters. Some of the games are action/arcade, but several are thinking games requiring you to solve problems. Looks very nice.

Adventures with the Atari by Jack Hardy ($15, 356 pages.) For all you adventure fans, this book teachs you how to make your own graphic adventure games and includes six adventure game listings to type in and study as well as play. The truly unique part is all the listings and instuctions are written in Atari BASIC, Atari MicroSoft BASIC and Atari PILOT!! What a gold mine for learning to translate. This book will be reviewed extensively next issue by Ruth Ellsworth, who has written in ACE an article on programming Adventure games with PILOT.

Survival on Planet X by Michael Orkin and Ed Bogas ($13, 151 pages). This is an adventure story with many short programs written in Atari BASIC; it teaches the basics of programming to beginners in a very painless way. Looks like a good introduction for a early teenager to learn how to use and program with the BASIC commands, including some fairly advanced graphics. A very nice book, interesting to read and you learn in spite of yourself!

Rainy Day Activities for the Atari by Nancy Mayer ($13, 156 pages). The forward to this book says you can turn your Atari "into a toy store that will provide fifty games of music, art, letters, numbers, and nonsense for young children."

Hardware Items of the Year

1. ATR 8000 with CP/M (SWP, 2500 Randol Mill Road-125, Arlington, TX 76011). This combination disk drive interface, printer buffer and interface and RS-232 port is a very low priced accessory for your Atari that opens new worlds. The basic unit allows you to use any "generic" disk drives and gives you a nice printer buffer. More advanced units allow you to use CP/M and there is even a 8088 board. It comes with programs allowing you to read and run almost any CP/M program from any CP/M computer, with all that wonderful business software and public domain programs.

2. Austin Franklin 80 column board. (Austin Franklin Assoc., 43 Grove St, Ayer, MA 01432) This 80 column board works very nicely, can be used with a RGB monitor (not for Atari Games), and has a cartridge to use with the ATR 8000. If you use CP/M, 80 columns are almost a must, and this is a nice one.

3. Mannesmann Taly Spirit 80 Printer. This new printer is an Epson work-alike with a big difference — square dots rather than round dots. It also has the best paper handling I've seen, with a very easy to use tractor feed as well as friction feed. The listings last month in the newsletter were done by this printer. This month, they will be the Epson-80. My Spirit broke— although you can easily back up the paper, if you pull the paper out the back, you can break the platen, and I did. The instruction manual doesn't tell you this, but the company did when I told them the problem! Otherwise, a very nice printer with very good print, and for only $399 retail, a good buy.

4. Brother 50 Daisy Wheel Typewriter with Brother IF-50 Computer interface. One of several new typewriter/computer printers factory designed for double duty. This is a nice inexpensive typewriter which works very well as a printer.

BUGS

(This game is appearing simultaneously in Page 6, a user group 'zine in England).

Bugs is a simple game which I hope will show that some good games can be written for your Atari even if you don't understand some of the the more complex programming routines. No player-missile graphics, no VBI routines, all of the action is achieved by POKEs to the screen in Antic mode 4. By level 9 the game gets to be quite fast.

You play the part of a ladybug who has to harvest a whole host of aphids by collecting them into one of four"nests"on each side of the screen. Dotted around the place are several mushroom looking things which, if you touch them, hatch out into more aphids. These are poisonous to a degree and will sap your strength but strength can be made up by killing the aphids by pushing them right into the nests. There are some aphids which are deadly (the orange ones) and touching these means instant death. If you are not quick in collecting the aphids, these will multiply. If your strength falls to zero, you die and you must have a certain strength ratio to progress to the next level, otherwise you must play the level again.

Enjoy the harvest but watch out for those killer bugs.

—Les Ellingham

STAN OCKERS

ALPHABET TRAIN

String operations on the Atari can be used to move data around in memory at machine level speeds. I've used this method for movement in a game called 'Alphabet Train' which is designed to train preschoolers in the positional aspect of letters in the alphabet.

The train and truck in the program consist of strings of characters. Parts of these strings are mapped onto the screen area. Motion is produced by changing the portion of the string being mapped. Movement of the boat in last month's program 'Cannibals' was done by using this method.

Another string is set aside as the player missile area. Vertical movement of players representing the grabber, pushers and selected letter is also done by using string operations.

The key to using these techniques is a table holding information about all variables. It is called the Variable Value Table and is built up as information is entered from the keyboard. Each time a new variable is encountered, eight bytes are set aside to hold the relavent information. For strings, this includes where to find the string, how long it is, and how much space was set aside when it was dimensioned.

The second and third of the eight bytes give the string location in the standard low byte, high byte order. This is not an absolute address, however, but is an offset from a particular spot in memory; The spot being the beginning of the area holding the string characters (STARP).

Since we want to 'fool' the Atari into thinking the string is located somewhere else, all that is necessary is to have the string dimensioned, (you can give it a DIM of 1). This will create the eight bytes. We must then know where the eight bytes are located so we can change them at will.

Locations 134 and 135 (decimal) hold a pointer to the variable value table (VVTP). If we arrange things so the first variables entered are ones upon which we want to operate, then we know the appropriate locations.

The first line entered (exclusive of REMs) should be a DIM statement with the string variables in the order we want them to appear in the table. Remember it is the order they are entered from the keyboard; so if we later decide we want another string and modify this first line, we must 'LIST' to disk or cassette and 'ENTER' the program again so the variable value table will be rebuilt just as if we were entering it from the keyboard for the first time.

A$ is the first variable found in the 'Train' program and is changed to point at the previously allocated Player Missile area. The offset is initially zero (it is at the start of the variable area), but is changed to the difference between the player part of the PM area and this position by poking numbers into the third and forth bytes of the eight representing variable A$ (VVTP + 2 & VVTP + 3). The size (5th and 6th) and dimesion (7th and 8th) are also changed to 512 bytes, the size of the player portion of the PM area, (see lines 350 - 390).

It should be possible to use the variable value table changing technique to move the character set into RAM but when I tried, it got all messed up so I just wrote a simple machine level user call to move one page. Since only 64 characters are available in Gr. 18 and require only 512 bytes, I used the unused portion at the beginning of the PM area and chose to have graphic symbols and capital letters.

PM images are assigned to strings with zero bytes at the ends so when vertical movement occurs, the previously drawn image will be wiped out. When a letter is selected, the grabber is lowered (Player #0), the letter blanked out from the train string and reconstructed as Player #1. It continues as player #1 until finally dropped on the truck where it is erased and entered into the proper truck string.

Vertical movement seems quite smooth although you might notice a slight jitter between the grabber and letter caused by successive string movements from Basic.

B$ is the second variable in the table and is changed in size to 261 bytes, (the size of the graphics 18 screen), and mapped over the screen area. Movement of images in this area is simply a matter of string manipulations. You are of course limited to multiples of a character size for movement but it doesn't seem too unrealistic in the case of the train and truck.

It should be pointed out the characters assigned to strings are not the characters appearing on the screen unless they are put on using PRINT statements. This is because the ATASCII order of characters is not the same as the order they appear in ROM. In a PRINT statement, a number is assigned based on the ATASCII order, (the one you get with ASC), and a conversion is made to reflect the order the character appears in ROM. Since we are mapping directly onto the screen, no conversion is made and allowance must be made when we choose characters.

I anticipated problems with jerkyness because Basic is so slow. I especially worried about the truck and train because they take three seperate string operations for each motion. I think you will agree the worry was in vain and that movement using strings is a valuable technique which can be used in place of machine routines.

PILOT RUTH

MORE STRINGS

by Ruth Ellsworth

In the hope of untangling beginners who find strings one of the most confusing concepts in programming, this is the first of two articles about using strings in PILOT. This article will give examples of the simple use of string variables, those cases in which strings may be thought of as slots in a toy shelf or files in cabinet depending upon the age of the user. Those interested in a more condensed review of the string function of PILOT may read my article in the December/January 1983 issue of the ACE Newsletter.

In ATARI PILOT any group of characters used as a unit is viewed as a string. One could think of string variables as slots built into a toy shelf. Each slot has a spe¬ific name, for example $ONE puts that label on a slot. Anything chosen could be placed in that particular slot, to find the thing in the slot titled $ONE, one looks into $ONE. In PILOT it is possible to actually look into all the string variable slots and see what they contain by using the DUMP command.

An activity our family enjoyed when we were new to strings was to write a simple program allowing the children to fill string variables, then to DUMP them. Such a program was more fun if the children took turns so they did not see the values given to each variable until the program was dumped:

```
T:Type a holiday
A:$HOLIDAY
T:Type an animal
A:$ANIMAL
```

Once we had begun to understand what a string variable was and how it worked, it was easy to begin to write programs using them. Mad Libs and Fractured Fairy Tales quickly became very popular. Also useful were form letters, but we quickly discovered that unless Aunt Martha, etc. were present, such programs were quite useless.

The first program below demonstrates a simple Thank You letter (very appropriate at Christmas time) which will dump to a printer. The difference between the screen dump and the printer dump is the use of the T: and WRITE:P, commands. The body of the letter could be enlarged to include as many lines as one desires. The only rule to be remembered is all T: commands in the letter must be replaced by WRITE:P, commands to print.

The second program is an example of a song between a Mad Lib and Fractured Fairy Tale. The *YES module allows one to rerun the story without changing variables. This program could also be dumped to the printer substutiting WRITE:P, commands for the T: commands in the *YES module.

Have fun experimenting with simple string variables over the Holidays. In February we should be ready to really "take off" with string variables by learning to combine them, divide them, and call them up randomly.

RUTH ELLSWORTH

```
0 ; by Ruth Ellsworth
10 R:AN ALL PURPOSE THANK YOU NOTE, JU
ST IN TIME FOR CHRISTMAS.
20 *MAIN
30 T:TYPE THE NAME OF YOUR BENEFACTOR
40 A:$NAME
50 T:TYPE THE NAME OF THE GIFT
60 A:$GIFT
70 T:TYPE YOUR NAME
80 A:$THANKFUL
90 T:}
100 T:DEAR $NAME
110 T:THANK YOU FOR $GIFT.
120 T:SINCERELY,
130 T:$THANKFUL
140 T:
150 T:
160 T:
170 T:IS YOUR NOTE CORRECT AND READY T
O PRINT?
180 A:
190 M:Y,N
200 JM:*YES,*NO
210 E:
220 *NO
230 T:}
240 VNEW:
250 J:*MAIN
260 E:
270 *YES
280 T:IS THE PRINTER TURNED ON?
290 A:
300 M:Y,N
310 JM:*ON,*OFF
311 E:
312 *OFF
313 PA:90
314 J:*ON
315 E:
316 *ON
320 WRITE:P,DEAR $NAME
330 WRITE:P,THANK YOU FOR $GIFT.
340 WRITE:P,SINCERELY,
350 WRITE:P,$THANKFUL
360 E:
10 R:JOLLY OLD SAINT NICHOLAS?
20 R:AN EXAMPLE OF A MAD LIB - FRACTUR
ED FAIRY TALE TYPE PROGRAM.
30 R:MUSIC COULD BE ADDED TO THIS ONE
TO MAKE IT A SILLY SONG IF DESIRED.
40 R:THIS TYPE OF PROGRAM COULD BE
CHANGED TO ALLOW THE STRING VARIABLES
50 R:TO BE ASSIGNED AT THE BEGINNING
OF THE PROGRAM.
```

```
60 R:SPECIFIC TYPES OF WORDS COULD BE
ASKED FOR IF DESIRED.
70 R:THE LINES MIGHT READ 10 T:TYPE A
NOUN
80 R:20 A:$NOUN
90 R:ETC.
100 R:THE A: LINES WOULD BE REMOVED
 FROM THE BODY OF THE PROGRAM BELOW,
 AND THE
110 R:MAD LIB WOULD RUN UNINTERRUPTED
 IN THAT EVENT.
120 *START
130 T:}
140 T:JOLLY OLD \
150 A:$ONE
160 A:$TWO
170 T:YOUR EAR THIS WAY!
180 T:DON'T YOU TELL A SINGLE \
190 A:$THREE
200 T:WHAT I'M GOING TO \
210 A:$FOUR
220 A:$FIVE
230 T:IS COMING SOON
240 T:NOW, YOU DEAR OLD \
250 A:$SIX
260 A:$SEVEN
270 T:WHISPER WHAT YOU'LL \
280 A:$EIGHT
290 T:TO ME;
300 A:$NINE
310 T:ME IF YOU CAN.
320 T:
330 T:
340 T:
350 T:WOULD YOU LIKE TO RERUN THIS SON
G LIKE IT IS?
360 A:
370 M:Y,N
380 JM:*YES,*NO
390 E:
400 *NO
410 VNEW
420 J:*START
430 E:
440 *YES
450 T:}
460 T:JOLLY OLD $ONE,
470 T:$TWO YOUR EAR THIS WAY!
480 T:DON'T YOU TELL A SINGLE $THREE
490 T:WHAT I'M GOING TO $FOUR
500 T:$FIVE IS COMING SOON;
510 T:NOW, YOU DEAR OLD $SIX
520 T:$SEVEN WHAT YOU'LL $EIGHT TO ME;
530 T:$NINE ME IF YOU CAN.
540 E:
```

ALPHABET by Stan Ockers

```
0 REM Printed on a Mannesmann Tally Sp
irit-80 printer with square dots
1 REM ******************************
2 REM *Atari Computer Enthusiasts  *
3 REM *     3662 Vine Maple Dr     *
4 REM *        EUGENE, OR 97405    *
5 REM *            $12 YEAR        *
6 REM *           DEC 83/JAN 84
7 REM ******************************
100 REM ****************************
110 REM **       ALPHABET TRAIN    **
120 REM ** STAN OCKERS 11/83       **
130 REM ****************************
140 REM
142 REM ** DEFINE STRINGS **
148 GRAPHICS 18:POSITION 5,5:? #6;"AlP
hAbEt":POSITION 7,7:? #6;"tRaIn":POSIT
ION 2,10:? #6;"INITIALIZING .."
150 DIM A$(1),SCR$(1),B$(12),TRN1$(180
),TRN2$(180),TRN3$(180),TRK1$(28),TRK2
$(28),TRK3$(28),W$(6)
160 DIM L$(10):L$(10)=CHR$(0):L$(1,1)=
CHR$(0):L$(2)=L$:DIM BL$(10):BL$=L$
170 DIM P$(8):RESTORE 180:FOR J=1 TO 8
:READ A:P$(J,J)=CHR$(A):NEXT J
180 DATA 1,3,7,255,255,7,3,1
190 DIM R$(8):RESTORE 200:FOR J=1 TO 8
:READ A:R$(J,J)=CHR$(A):NEXT J
200 DATA 128,192,224,255,255,224,192,1
28
210 RESTORE 220:FOR J=1 TO 12:READ A:B
$(J,J)=CHR$(A):NEXT J
220 DATA 24,36,66,129,129,129,129,129,
129,129,66,0
230 TRN1$(180)=" ":RESTORE 232:FOR J=1
 TO 24:READ A:TRN1$(J,J)=CHR$(A):NEXT
J:TRN1$(25)=TRN1$
232 DATA 77,84,84,0,84,84,205,212,212,
128,212,212,13,20,20,0,20,20,141,148,1
48,128,148,148
240 TRN2$(180)=" ":RESTORE 242:FOR J=1
 TO 24:READ A:TRN2$(J,J)=CHR$(A):NEXT
J:TRN2$(25)=TRN2$
242 DATA 0,32,32,32,32,32,0,96,96,96,9
6,96,0,160,160,160,160,160,0,224,224,2
24,224,224
250 TRN3$(180)=CHR$(0):TRN3$(1,1)=CHR$
(0):TRN3$(2)=TRN3$:FOR J=10 TO 160 STE
P 6:TRN3$(J,J)=CHR$(J/6+95):NEXT J
260 RESTORE 262:FOR J=1 TO 28:READ A:T
RK1$(J,J)=CHR$(A):NEXT J
262 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,84,84,0,0,0,0,84,84,0,84,0,0,
264 RESTORE 266:FOR J=1 TO 28:READ A:T
RK2$(J,J)=CHR$(A):NEXT J
266 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,224,224,224,224,224,224,224,21,32,32,
32.32.0.
270 RESTORE 272:FOR J=1 TO 28:READ A:T
RK3$(J,J)=CHR$(A):NEXT J
272 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,32,13,22,0,0,
278 REM ** SET ASIDE PM AREA **
280 GRAPHICS 7:POKE 106,PEEK(106)-5:PM
=PEEK(106)+1:POKE 54279,PM:GRAPHICS 18
:GOSUB 910
290 RESTORE 300:FOR J=704 TO 712:READ
A:POKE J,A:NEXT J
300 DATA 32,255,18,82,50,255,34,52,57
310 POSITION 5,1:? #6;CHR$(142);CHR$(1
42);CHR$(142):POSITION 6,6:? #6;"-----
----------"
320 POSITION 0,10:? #6;"++++++++++++++
++++++";:REM ** ALL +'S IN INVERSE **
330 POKE 559,46:POKE 53277,3
340 REM ** MAP A$ ONTO PM AREA **
350 VVTP=PEEK(134)+PEEK(135)*256:STARP
=ADR(A$)
380 OFFS=PM*256+512-STARP:HI=INT(OFFS/
256):LO=OFFS-HI*256
390 POKE VVTP+2,LO:POKE VVTP+3,HI:POKE
 VVTP+5,2:POKE VVTP+6,0:POKE VVTP+7,2
392 REM ** MAP SCR$ ONTO SCREEN AREA *
*
400 SCR=PEEK(88)+256*PEEK(89):OFFS=SCR
-STARP:HI=INT(OFFS/256):LO=OFFS-HI*256
420 POKE VVTP+10,LO:POKE VVTP+11,HI
430 POKE VVTP+12,5:POKE VVTP+13,1:POKE
 VVTP+14,5:POKE VVTP+15,1
440 Y=60:Z=80:X=50
450 A$(Y,Y+11)=B$:POKE 53248,Z:FOR J=1
 TO 59:A$(J,J)=CHR$(24):NEXT J
460 A$(280,287)=P$:XP=64:POKE 53250,XP
470 A$(408,415)=R$:XR=158:POKE 53251,X
R
478 REM ** TRUCK COMES IN FROM RIGHT *
*
480 FOR J=1 TO 15:XT=J:GOSUB 1050:SOUN
D 0,100,4,XT/2:FOR K=1 TO 20:NEXT K:NE
XT J:POKE 77,0
488 REM ** PRINT WORD **
490 GOSUB 1010:POSITION 10-LEN(W$)/2,1
1:? #6;W$
498 REM ** LOOP TO MOVE TRAIN **
500 S=STICK(0):IF S=7 AND X>6 THEN X=X
-1:DIR=-1
510 IF S=11 AND X<156 THEN X=X+1:DIR=1
520 IF S<15 THEN 560
530 IF INT(X/6)=X/6 THEN 560
540 IF DIR=-1 THEN X=X-1
550 IF DIR=1 THEN X=X+1
558 REM ** MOVE TRAIN **
560 SCR$(181,200)=TRN1$(X,X+19)
570 SCR$(161,180)=TRN2$(X,X+19)
578 REM ** SOUND IF TRAIN MOVING **
580 SCR$(141,160)=TRN3$(X,X+19)
590 IF X=SX THEN SOUND 0,0,0,0
600 IF X<>SX THEN SOUND 0,100,4,6:SX=X
610 IF STRIG(0)=1 THEN 500
618 REM ** PICK UP LETTER **
620 C=ASC(TRN3$(X+4)):IF C=0 THEN 500
630 FOR J=1 TO 11:Y=Y+1:A$(Y,Y+11)=B$:
SOUND 0,(Y+100)*(Y/3=INT(Y/3)),10,6:FO
R K=1 TO 20:NEXT K:NEXT J
640 CS=57344+256+8*X/6:FOR J=1 TO 8:L$
(J+1,J+1)=CHR$(PEEK(CS+J-1)):NEXT J
650 A$(129+Y,139+Y)=L$:POKE 53249,Z
660 P=4+X:TRN3$(P,P)=CHR$(0):SCR$(141,
160)=TRN3$(X,X+19)
670 FOR J=1 TO 50:Y=Y-1:A$(Y,Y+11)=B$:
A$(129+Y,139+Y)=L$:SOUND 0,(Y+100)*(Y/
3=INT(Y/3)),10,6:FOR K=1 TO 10
680 NEXT K:NEXT J:SOUND 0,0,0,0:FOR J=
1 TO 8:XP=XP+1:POKE 53250,XP:NEXT J
688 REM ** IF LETTER NOT CORRECT DROP
IT **
690 IF C-32<>ASC(W$(LTR+1)) THEN GOSUB
 820
700 FOR J=1 TO 32+8*LTR:Z=Z+1:XP=XP+1:
POKE 53250,XP:POKE 53249,Z
710 SOUND 0,Y+100*(XP/3=INT(XP/3)),10,
6:NEXT J:LTR=LTR+1
720 FOR J=1 TO 15:Y=Y+1:A$(129+Y,139+Y
)=L$:SOUND 0,Y+100,10,6:NEXT J:A$(129+
Y,139+Y)=BL$
730 P=16+LTR:TRK3$(P,P)=CHR$(X/6+96):S
CR$(67,80)=TRK3$(XT,XT+13):POSITION (Z
-48)/8,1:? #6;CHR$(142)
740 FOR J=1 TO 30:SOUND 0,100+100*RND(
0),8,(30-J)/3:NEXT J
750 XP=XP-1:POKE 53250,XP:SOUND 0,Y+10
0*(XP/3=INT(XP/3)),10,6:IF XP>64 THEN
750
760 Z=80:POKE 53249,Z:Y=Y-15
770 FOR J=1 TO 39:Y=Y+1:A$(Y,Y+11)=B$:
SOUND 0,Y+100,10,6:NEXT J
780 SOUND 0,0,0,0:IF LTR<LEN(W$) THEN
500
788 REM ** WORD CORRECT - MOVE TRUCK O
UT **
790 FOR J=15 TO 1 STEP -1:XT=J:GOSUB 1
050:SOUND 0,100,4,XT/2:FOR K=1 TO 20:N
EXT K:NEXT J
795 GOSUB 2040
800 POSITION 7,11:? #6;"        ":FOR J=
10 TO 160 STEP 6:TRN3$(J,J)=CHR$(J/6+9
5):NEXT J
810 LTR=0:FOR J=17 TO 22:TRK3$(J,J)=CH
R$(0):NEXT J:POSITION 8,1:? #6;"
":GOTO 480
818 REM ** ROUTINE FOR PUTTING INCORRE
CT LETTER BACK **
```

```
820 FOR J=1 TO 14:SOUND 0,50+50*(XP/3=
INT(XP/3)),10,6
830 Z=Z+1:XP=XP+1:XR=XR-4:POKE 53249,Z
:POKE 53250,XP:POKE 53251,XR:NEXT J:SO
UND 0,0,0,0
840 FOR J=1 TO 14:Z=Z-1:XP=XP-1:XR=XR-
1:POKE 53249,Z:POKE 53250,XP:POKE 5325
1,XR:NEXT J
850 FOR J=1 TO 8:XP=XP-1:XR=XR+2:POKE
53250,XP:POKE 53251,XR:Y=Y+1:A$(129+Y,
139+Y)=L$:A$(Y,Y+11)=B$:NEXT J
860 FOR J=1 TO 27:XR=XR+2:POKE 53251,X
R:Y=Y+1:A$(129+Y,139+Y)=L$:A$(Y,Y+11)=
B$:SOUND 0,Y+100,10,8:NEXT J
870 FOR J=1 TO 15:SOUND 0,Y+100,10,8:Y
=Y+1:A$(Y,Y+11)=B$:A$(129+Y,139+Y)=L$:
NEXT J:TRN3$(P,P)=CHR$(C)
880 FOR J=1 TO 30:SOUND 0,100+100*RND(
0),8,(30-J)/3:NEXT J
890 SCR$(141,160)=TRN3$(X,X+19):A$(129
+Y,139+Y)=BL$:FOR J=1 TO 11:Y=Y-1:A$(Y
,Y+11)=B$:NEXT J
900 SOUND 0,0,0,0:GOTO 500
908 REM ** MACHINE LEVEL - MOVE CHARAC
TER SET **
910 POKE 756,PM
920 DIM MV$(23):RESTORE 930:FOR J=1 TO
 23:READ A:MV$(J,J)=CHR$(A):NEXT J
930 DATA 104,104,133,204,104,133,203,1
04,133,206,104,133,205,160,0,177,203,1
45,205,136,208,249,96
940 CS=57344+512:NCS=PM*256:A=USR(ADR(
MV$),CS,NCS)
950 FOR J=NCS TO NCS+7:POKE J,0:NEXT J
960 CS=57344+256:NCS=PM*256+256:A=USR(
ADR(MV$),CS,NCS)
970 FOR J=NCS TO NCS+7:POKE J,255:NEXT
 J
980 NCS=PM*256:RESTORE 990:FOR J=160 T
O 167:READ A:POKE NCS+J,A:NEXT J
990 DATA 0,0,60,102,219,219,102,60
1000 POKE 756,PM:RETURN
1008 REM ** PICK A WORD SUBROUTINE **
1010 RESTORE 1020:FOR J=1 TO INT(RND(0
)*10)+1:READ W$:NEXT J:RETURN
1020 DATA DOG,CAT,BIRD,HAND,COW,HORSE,
PET,FROG,FLY,ARM
1030 DATA HAIR,TOE,FINGER,LEG,FACE,EAR
,MOUTH,BAT,WORK,PLAY
1040 DATA WAGON,BIKE,MILK,WATER,JUICE,
BREAD,ANT,GOAT,GHOST,FLAT
1048 REM ** PRINT TRUCK SUBROUTINE **
1050 SCR$(107,120)=TRK1$(XT,XT+13):SCR
$(87,100)=TRK2$(XT,XT+13):SCR$(67,80)=
TRK3$(XT,XT+13):RETURN
2030 REM ** TUNE SUBROUTINE **
2032 DATA 91,6,121,1,91,2,121,1,91,2,8
1,1,72,8,91,8,68,6,68,1,91,4,81,4,72,8
,0,4
2034 DATA 91,6,121,1,91,2,121,1,91,2,8
1,1,72,8,91,4,72,2,72,1,72,4,81,4,81,4
,72,4,81,8
2040 RESTORE 2032:FOR J=1 TO 29:READ P
,T:IF P=0 THEN FOR K=1 TO 20*T:NEXT K:
GOTO 2070
```

BUGS by Les Ellingham

Printed with an Epson MX-80

```
0 REM ********************************
1 REM *             BUGS             *
2 REM *       by Les Ellingham       *
3 REM *       Stafford, England      *
4 REM *     for PAGE 6 magazine.     *
5 REM *  Submitted to ACE library    *
6 REM *  in thanks for many fine     *
7 REM *           programs.          *
8 REM ********************************
9 REM  ACE Newsletter, 3662 Vine Map
le Eugene, OR 97405
10 TRAP 10000:GOTO 1000
19 REM ** DELAY SUBROUTINE **
20 FOR D=1 TO DL:NEXT D:RETURN
29 REM ** UP-DATE SCORE ETC **
30 IF SC>CSC THEN CSC=SC:SCORE=CSC+BSC
35 STR=INT(SC)/3:IF STR<0 THEN STR=0
40 POSITION 20,0:? "STR";INT(STR);
"  ":POSITION 29,0:? "SCORE";SCOR
E;
45 IF STR<=0 THEN POP :GOTO 600
50 POKE 77,0:RETURN
55 POKE 77,0:RETURN
59 REM ** CHECK FOR COLLISIONS ETC **
60 M=Z2:M2=Z2-Z
65 IF PEEK(M+M2)=3 THEN 150
70 IF PEEK(M+M2)=4 OR PEEK(M+M2)=5 OR
PEEK(M+M2)=7 THEN 95
75 POKE M,0:POKE M+M2,4
80 SOUND 1,10,8,6:SOUND 1,0,0,0:RETURN

95 FOR I=0 TO 7:M2=MPOS(I)
105 IF PEEK(M+M2)=0 THEN POP :GOTO 75
110 NEXT I
115 POKE Z,0:POKE M,6:GOSUB 815
120 BUGS=BUGS-1:SC=SC+10:SCORE=SCORE+2
:CSC=CSC+2:POKE 20,0:POKE 19,0:IF BUGS
<=0 THEN POP :GOTO 500
125 M=0:M2=0:GOSUB 30
130 RETURN
150 POKE M+M2,4:BUGS=BUGS+1:TS=TS-1:SC
=SC-((LV*6+6)-TS)*4:GOSUB 30:GOTO 60
199 REM ** MAIN GAME ROUTINE **
200 S=STICK(0):IF S=15 THEN Z2=Z+MPOS(
INT(RND(0)*8)):FOR D=1 TO 72-LV*8:NEXT
 D:GOTO 215
210 Z2=Z+ST(S)
215 IF PEEK(Z2)=5 THEN 200
220 IF PEEK(Z2)=3 THEN GOSUB 150
225 IF PEEK(Z2)=4 THEN GOSUB 60
230 IF PEEK(Z2)=7 THEN GOTO 600
235 IF PEEK(19)>3 THEN GOSUB 900
240 FOR D=1 TO 45-LV*5:NEXT D
245 POKE Z,0:Z=Z2:POKE Z,6
250 SOUND 0,12,10,4:SOUND 0,0,0,0
255 GOTO 200
499 REM ** LEVELS AND BONUSES **
500 IF SC<CSC THEN 580
520 GRAPHICS 17:POSITION 6,9:? #6;"LEV
EL ";LV
525 FOR I=1 TO 20-((LV*6+6)-TS):POSITI
ON I-1,11:? #6;"*":BONUS=BONUS+5
530 V=2:GOSUB 810:V=1
535 DL=10:GOSUB 20
540 POSITION 6,13:? #6;"BONUS ";BONUS
545 NEXT I:BSC=BSC+BONUS
550 DL=500:GOSUB 20
555 LV=LV+1:IF LV>9 THEN LV=9
560 ? #6;CHR$(125):POSITION 6,9:? #6;"
LEVEL ";LV:SOUND 0,255,10,8:SOUND 1,25
4,10,8
565 FOR I=0 TO 255:POKE 708,I:NEXT I
575 DL=500:GOSUB 20:GOSUB 820
580 BONUS=0:GRAPHICS 0:POKE 755,0:POKE
 710,0:GOTO 1260
599 REM ** END OF GAME ROUTINE **
600 FOR I=255 TO 0 STEP -5:SOUND 0,255
-I,10,8:FOR D=1 TO 5:NEXT D:POKE Z,I:N
EXT I
610 FOR I=1 TO 3:GOSUB 815:NEXT I
620 GRAPHICS 18:POSITION 5,0:? #6;"GAM
E OVER"
625 IF SCORE>HI THEN HI=SCORE
630 POSITION 2,2:? #6;">>>>>>>><<<<<<<
<"
635 POSITION 5,4:? #6;"SCORE  ";SCORE
640 POSITION 5,6:? #6;"HI-SC  ";HI
645 POSITION 2,8:? #6;">>>>>>>><<<<<<<
<"

650 POSITION 3,10:? #6;"press start
 to          play again"
655 POKE 711,PEEK(20)
660 IF PEEK(53279)<>6 THEN 655
665 GRAPHICS 0:POKE 710,0:POKE 755,0:S
C=10:CSC=0:BSC=0:LV=1:GOTO 1260
800 REM ** SOUNDS ETC **
810 FOR S=14 TO 0 STEP -V:SOUND 0,S*10
,12,S:NEXT S
```

→

Print Atari Graphic Charactors by Greg Menke

```
0 REM Printed on a Mannesmann Tally Sp
irit-80 printer with square dots
1 REM *****************************
2 REM *Atari Computer Enthusiasts  *
3 REM *     3662 Vine Maple Dr     *
4 REM *      EUGENE, OR 97405      *
5 REM *          $12 YEAR          *
6 REM *        DEC 83/JAN 84
7 REM *****************************
10 REM Graphics Characters
20 REM on the MX-80 with Graphtrax.
30 REM
40 REM By Greg Menke.  11/8/83
50 REM
60 REM (301)972-8324
62 REM
64 REM
66 REM
70 GRAPHICS 0:DIM A$(20),B$(9),C$(4),C
HAR(8,8),A(480),TITLE$(120)
72 SIZE=FRE(0)-500:DIM BUFF$(SIZE):BUF
F$(1)=" ♥":BUFF$(SIZE)=" ♥":BUFF$(2)=BU
FF$:SECTORS=INT(SIZE/125)
75 FOR G=0 TO 8:FOR H=0 TO 8:CHAR(H,G)
=0:A(G)=0:NEXT H:NEXT G:GOSUB 2000:FOR

G=0 TO 480:A(G)=0:NEXT G
80 ? :? :? "Enter filename for print."
:? "(You have ";SIZE;" free buffer byt
e
s,":? " ";SECTORS;" Sectors)."
90 INPUT A$:TRAP 100:GOTO 110
100 ? :? "I can't find ";A$:GOTO 80
110 TRAP 40000:OPEN #1,4,0,A$:OPEN #2,
8,0,"P:"
112 ? :? :? "Do you want the modified
SPACE":? "character (Y/N) ";:INPUT A$
115 IF A$<>"Y" AND A$<>"N" THEN 112
120 GRAPHICS 0:? "Do you want a screen
 echo of printout   (Y/N) ";:INPUT C$
122 IF C$<>"Y" AND C$<>"N" THEN 120
125 ? :? "Enter Printout title.":? "(R
ETURN only if none.)":? :INPUT TITLE$
127 IF LEN(TITLE$)>0 THEN ? #2:? #2;TI
TLE$:? #2:? #2:? #2:? #2
130 ? :? "Enter new character set #.":
? "(RETURN only for no change.)":INPUT

A$
135 IF LEN(A$)>0 THEN POKE 756,VAL(A$)
140 GRAPHICS 0:? #2;CHR$(27);"<"
141 BIN=1:TRAP 147:NO=0:FLAG=0:L=LEN(B
UFF$)
142 GET #1,A:BUFF$(BIN,BIN)=CHR$(A):BI
N=BIN+1:? "E";CHR$(A);
143 IF BIN=L THEN POKE 559,34:CLOSE #2
:CLOSE #1:? :? "KKKFile too
large.....":? :TRAP 40000:? :?
:GOTO 540
145 GOTO 142
147 GRAPHICS 0:NO=BIN:BIN=1:CLOSE #1:I
F C$="N" THEN POKE 559,0
148 IF LEN(A$)>0 THEN POKE 756,VAL(A$)
150 FOR BYTE=0 TO 1 STEP 0:INVERSE=0
155 A=ASC(BUFF$(BIN,BIN)):POKE 53279,0
:POKE 53279,8:IF C$="Y" THEN ? CHR$(27
)
;CHR$(A);
157 BIN=BIN+1:TRAP 510:IF (BIN-1)=NO T
HEN FLAG=1:GOTO 510
170 IF A=155 THEN X=0:Y=OFFSET:GOSUB 2
15:? #2:GOTO 500
180 IF A>127 THEN A=A-128:INVERSE=1
190 IF A>31 AND A<96 THEN A=A-32:GOTO
205
200 IF A<32 THEN A=A+64
205 IF A=0 AND A$="Y" AND  NOT INVERSE
 THEN GOSUB 1700:GOTO 500
210 LOC=(PEEK(756)*256)+A*8:GOSUB 1000
:GOTO 500
211 REM
215 IF OFFSET>255 THEN X=1:Y=OFFSET-25
6
217 ? #2;CHR$(27);"K";CHR$(Y);CHR$(X);
220 FOR G=1 TO OFFSET:? #2;CHR$(A(G));
:NEXT G:OFFSET=0:RETURN
500 IF OFFSET=480 THEN X=0:Y=OFFSET:GO
SUB 215:? #2
501 NEXT BYTE
502 REM
503 REM
510 IF OFFSET>0 THEN X=0:Y=OFFSET:GOSU
B 215
515 GRAPHICS 0:TRAP 40000:CLOSE #2:CLO
SE #1:? "▣":FOR G=1 TO 20:LPRINT :NEX
T
 G
520 IF  NOT FLAG THEN GRAPHICS 0:? :?
:? :? "Unexpected error :";PEEK(195):G
O
TO 540
530 ? :? :? "End of file. Printout com
plete.":? :?
540 ? "Press :":? :? "START to Re-
run.":? "SELECT to end."
550 IF PEEK(53279)=6 THEN CLR :RUN
560 IF PEEK(53279)=5 THEN ? :CLR :END
570 GOTO 550
995 REM
996 REM
997 REM
1000 REM CONVERT TO BINARY
1010 FOR G=0 TO 7:A=PEEK(LOC+G)
1020 C=128:FOR H=1 TO 8:CHAR(G+1,H)=0
1030 IF (A-C)>=0 THEN CHAR(G+1,H)=1:A=
A-C
1040 C=C/2:NEXT H:NEXT G
1050 IF INVERSE THEN FOR G=1 TO 8:FOR
H=1 TO 8:CHAR(G,H)=ABS(CHAR(G,H)-1):NE
X
T H:NEXT G
1070 REM CONVERT BACK TO DECIMAL
1080 FOR G=1 TO 8:C=128:NUM=0:FOR H=1
TO 8
1090 IF CHAR(H,G)=1 THEN NUM=NUM+C
1100 C=C/2:NEXT H:OFFSET=OFFSET+1:A(OF
FSET)=NUM:NEXT G
1110 RETURN
1700 REM MAKE SPECIAL SPACE CHARACTER
1710 B$="▚▀▀▀▀▀▀▀▀♥":LOC=ADR(B$)
1720 GOSUB 1010:RETURN
1997 REM
1998 REM
1999 REM
2000 REM DISK DIRECTORY
2010 OPEN #1,6,0,"D:*.*":TRAP 2030
2020 GET #1,A:? CHR$(A);:GOTO 2020
2030 TRAP 40000:CLOSE #1:RETURN
2050 SOUND 0,P,10,8:SOUND 1,2*P,10,6:F
OR L=1 TO 12*T
2060 NEXT L:SOUND 0,0,0,0:SOUND 1,0,0,
0
2070 NEXT J:RETURN
```

BOOTBUILD con't from Nov ACE

```
            0100 ; LIST#D:AUTBUILD
            0110 ;
            0120 ; ROUTING TO BOOT IN THE RS232 INTERFACE
            0130 ; THEN EXECUTE A BASIC STATEMENT
            0140 ; E.G., RUN"D:MENU
            0150 ;
0000        0160 DUMMY  =  $0000
E459        0170 SIO    =  $E459
F3E4        0180 SCRNRD =  $F3E4
000C        0200        *= $3800
3800 A950   0210        LDA #$50
3802 8D0003 0220        STA $0300      SET UP DCB TO
3805 A901   0230        LDA #1         INPUT "R" HANDLER
3807 8D0103 0240        STA $0301      FROM 850 INTERFACE
380A A93F   0250        LDA #$3F
380C 8D0203 0260        STA $0302
380F A940   0270        LDA #$40
3811 8D0303 0280        STA $0303
3814 A905   0290        LDA #5
3816 8D0503 0300        STA $0305
3819 8D0603 0310        STA $0306
381C A900   0320        LDA #0
381E 8D0403 0330        STA $0304
3821 8D0903 0340        STA $0309
3824 8D0A03 0350        STA $030A
3827 8D0B03 0360        STA $030B
382A A90C   0370        LDA #$0C
382C 8D0803 0380        STA $0308
382F 2059E4 0390        JSR SIO        TO INPUT "R" HANDLER
            0400 ;
3832 1003   0410        BPL IO.OK
3834 4C5738 0420        JMP LDBSC      IF CAN'T INPUT
            0430 ;
3837 A20B   0440 IO.OK  LDX #$0B       SET UP INPUT BUFFER
3839 BD0005 0450 LOOP   LDA $0500,X
383C 9D0003 0460        STA $0300,X
383F CA     0470        DEX
3840 10F7   0480        BPL LOOP
3842 2059E4 0490        JSR SIO        TO BOOT IN "R"
            0500 ;
3845 3010   0510        BMI LDBSC      IF BOOT FAILED
            0520 ;
3847 200605 0530        JSR $0506      TO INIT. RS232
384A A50C   0540        LDA DOSINI     TO CREATE IND. JSR
384C 8D5538 0550        STA INDJSR+1
384F A50D   0560        LDA DOSINI+1
3851 8D5638 0570        STA INDJSR+2
3854 200000 0580 INDJSR JSR DUMMY
            0590 ;
3857 A967   0600 LDBSC  LDA #SCRVEC    SET UP ADDRESS
3859 8D2103 0610        STA $0321      TO "STEAL" SCREEN EDITOR
385C A938   0620        LDA #$38       VECTOR TABLE
385E 8D2203 0630        STA $0322
            0640 ;
```

```
3861 A900   0650        LDA #$0        DUMMY VARIABLE FOR
3863 8DD638 0660        STA CMDLEN     LENGTH OF BASIC COMMAND
3866 60     0670        RTS            RETURN TO DOS
            0680 ;
0067        0690 SCRVEC =  *-$3800
3867 FBF3   0700        .DBYTE  $FBF3 COPY OF SCREEN EDITOR
3869 33F6   0710        .DBYTE  $33F6 VECTOR TABLE, EXCEPT
386B 7638   0720        .WORD   SCRNWT WITH NEW SCREEN WRITE
386D A3F6   0730        .DBYTE  $A3F6
386F 33F6   0740        .DBYTE  $33F6
3871 3CF6   0750        .DBYTE  $3CF6
3873 4CE4F3 0760        JMP SCRNRD
            0770 ;
3876        0780 SCRNWT =  *
3876 00     0790        .BYTE    $0
3877 ACD638 0800        LDY CMDLEN     NEW SCREEN WRITE ROUTINE
387A F009   0810        BEQ ENDBAS
387C B99138 0820        LDA BASCMD-1,Y WRITES THE STORED BASIC
387F CED638 0830        DEC CMDLEN     COMMAND TO THE SCREEN
3882 A001   0840        LDY #1
3884 60     0850        RTS
            0860 ;
3885 8C2103 0870 ENDBAS STY $0321      SETS UP VALUES FOR
3888 A9E4   0880        LDA #$E4       NORMAL RETURN
388A 8D2203 0890        STA $0322
388D A99B   0900        LDA #$9B
388F A001   0910        LDA #1
3891 60     0920        RTS
            0930 ;
3892        0940 BASCMD =  *           STORAGE RESERVED FOR
3892        0950        *= *+$44       BASIC COMMAND
38D6        0960 CMDLEN =  *           LENGTH OF BASIC COMMAND
38D6        0970        .END
```

```
815 FOR S=14 TO 0 STEP -V:SOUND 0,150-
S*10,12,S:NEXT S:SOUND 0,0,0,0:RETURN
820 SOUND 0,0,0,0:SOUND 1,0,0,0:RETURN

899 REM ** ADD NEW KILLER BUG **
900 X=SCREEN+INT(RND(0)*799)+40
910 IF PEEK(X)<>0 THEN 900
920 POKE X,7
930 POKE 19,0:POKE 20,0:RETURN
1000 REM ** TITLES AND INITIALISE **
1005 GRAPHICS 18:V=1
1010 FOR I=1 TO 17 STEP 2:POSITION I,0
:? #6;"==":POSITION I,11:? #6;"==":NEX
T I
1020 FOR I=1 TO 8:POSITION I,2:? #6;"B-
":POSITION I-1,2:? #6;" ":POSITION 19-
I,2:? #6;"s":POSITION 20-I,2:? #6;" "
1025 GOSUB 80:NEXT I
1029 DL=500:GOSUB 20
1030 POSITION 9,2:? #6;"Ug"
1035 GOSUB 810
1040 POSITION 9,5:? #6;"by":POSITION 4
,7:? #6;"les ellingham"
1045 GOSUB 1200:DL=750:GOSUB 20
1050 POSITION 2,5:? #6;"select lev
el ";LV
1055 POSITION 1,7:? #6;"press start
 to play"
1060 IF PEEK(53279)=5 THEN GOSUB 815:L
V=LV+1:IF LV>9 THEN LV=1:POSITION 15,5
:? #6;LV
1065 IF PEEK(53279)=6 THEN 1250
1080 P=PEEK(708):POKE 708,PEEK(710):PO
KE 710,P
1090 DL=250:GOSUB 20
1100 GOTO 1060
1200 REM
1210 DIM ST(14),MPOS(7)
1215 LV=1:SC=10:V=1
1220 FOR I=1 TO 14:READ A:ST(I)=A:NEXT
 I
1225 DATA 0,0,0,0,41,-39,1,0,39,-41,-1
,0,40,-40
1230 FOR I=0 TO 7:READ A:MPOS(I)=A:NEX
T I
1235 DATA 1,-1,39,-39,40,-40,41,-41
1245 RETURN
1250 POKE 106,PEEK(106)-5:GRAPHICS 0:P
OKE 710,0:POKE 755,0
1255 GOSUB 2500
1260 M=0:M2=0:Z=0:Z2=0:BUGS=LV*3:TS=LV
*6+6
1265 POKE 756,CS/256:GOSUB 2000
1270 POKE 710,10
1275 SCREEN=PEEK(88)+256*PEEK(89)
1280 GOSUB 1500:GOSUB 30:POKE 19,0:POK
E 20,0:GOTO 200
1499 REM ** DRAW SCREEN **
1500 FOR I=40 TO 79:POKE SCREEN+I,5:NE
XT I
1510 FOR I=80 TO 839 STEP 40:POKE SCRE
EN+I,5:POKE SCREEN+I+39,5:NEXT I
1520 FOR I=840 TO 879:POKE SCREEN+I,5:
NEXT I
1545 RESTORE 1560
1550 FOR I=1 TO 5:READ A:POKE SCREEN+8
85+I,A:NEXT I
1560 DATA 44,37,54,37,44
1570 POKE SCREEN+892,LV+16
1600 REM
1610 FOR I=1 TO LV*6+6
1620 X=SCREEN+INT(RND(0)*799)+40
1625 IF PEEK(X)=5 THEN 1620
1630 POKE X,3:NEXT I
1640 RESTORE 1650:FOR I=1 TO 24:READ A
:POKE SCREEN+A,5:NEXT I
1650 DATA 361,362,363,441,442,443,98,1
00,138,140,178,180,476,477,478,556,557
,558,740,742,780,782,820,822
1700 FOR I=1 TO LV*3
1710 X=SCREEN+INT(RND(0)*799)+40
1720 IF PEEK(X)=5 OR PEEK(X)=4 THEN 17
10
1730 POKE X,4:NEXT I
1740 Z=SCREEN+500:POKE Z,6
1750 X=SCREEN+INT(RND(0)*799)+40
1760 IF PEEK(X)=5 OR PEEK(X)=4 OR PEEK
(X)=3 THEN 1750
1770 POKE X,7
1790 RETURN
2000 REM * NEW DISPLAY LIST *
2005 POKE 559,0
2010 DL=PEEK(560)+256*PEEK(561)
2020 POKE DL+3,70:POKE DL+6,6
2030 FOR I=7 TO 27:POKE DL+I,4:NEXT I:
POKE DL+28,6:POKE DL+29,65
2040 POKE DL+30,PEEK(560):POKE DL+31,P
EEK(561)
2045 POKE 559,34
2050 RETURN
2500 REM * CHANGE CHARACTER SET *
2505 RESTORE 2530
2515 CS=(PEEK(106)+1)*256
2520 DIM MC$(32):FOR I=1 TO 32:READ A:
MC$(I,I)=CHR$(A):NEXT I:A=USR(ADR(MC$)
,57344,CS)
2530 DATA 104,104,133,204,104,133,203,
104,133,206,104,133,205,162,4
2535 DATA 160,0,177,203,145,205,136,20
8,249,230,204,230,206,202,208,240,96
2550 RESTORE 2580
2560 READ A:IF A=-1 THEN RETURN
2570 FOR J=0 TO 7:READ B:POKE CS+A*8+J
,B:NEXT J:GOTO 2560
2580 DATA 6,0,20,65,213,81,213,60,20
2585 DATA 4,130,40,170,170,170,40,40,1
30
2590 DATA 3,60,255,255,169,40,40,40,17
0
2592 DATA 7,65,20,85,85,85,20,20,65
2595 DATA -1
10000 GRAPHICS 0:? PEEK(195);" AT ";PE
EK(186)+256*PEEK(187)
```

Jerry White Sends Greetings

```
1 REM SEASONS GREETING TO MY FRIENDS I
N ACE- BY JERRY WHITE
100 GRAPHICS 0:POKE 82,10:POKE 83,39:P
OKE 752,1:POKE 764,255
110 SETCOLOR 2,13,4:SETCOLOR 4,4,2
120 POSITION 13,4:? "HAPPY HOLIDAYS":?
 :?
130 ?  ████   ████  ██████"
140 ? ██  ██  █  ██ █    █"
150 ? █    █ ██     █"
160 ? ██████ ██     ████"
170 ? █    █ ██     █"
180 ? █    █  █  ██ █    █"
190 ? █    █  ████  ██████"
200 ? :? :? "  from Jerry White"
210 COLOR 160:PLOT 0,0:DRAWTO 0,23:DRA
WTO 39,23:DRAWTO 39,0:DRAWTO 0,0
220 IF PEEK(764)=255 AND PEEK(53279)=7
 THEN POKE 709,PEEK(20):GOTO 220
230 POKE 764,255:GRAPHICS 0:POKE 82,2:
? :? "BASIC":? "IS";:END
```

TINYTEXT MK II by Bill Hardwick

```
0 REM ******************************
1 REM *                            *
2 REM *      TINY TEXT MARK II     *
3 REM *                            *
4 REM *      original  program     *
5 REM *  by Stan Ockers & Jim Carr *
6 REM *                            *
7 REM *  modified by Bill Hardwick *
8 REM *                            *
9 REM ******************************
11 REM ACE NEWSLETTER 3662 VINE MAPLE
EUGENE OR 97405
12 CLR :POKE 8,0:GRAPHICS 0:POKE 712,1
48:POKE 752,1:? CHR$(125):DIM PL$(1),S
P$(40):FOR I=1 TO 40:SP$(I,I)=" ":NEXT
 I
13 DIM C$(50),L$(1500),LP$(100),D$(256
)
14 DIM S$(45),I$(120),A$(128):SIZ=FRE(
0)-50:DIM T$(SIZ):FOR I=1 TO 45:READ A
:S$(I)=CHR$(A):NEXT I
20 DATA 104,104,133,204,104,133,203,10
4,133,206,104,133,205,104,104,168,162,
0,161,203,145,203,198,203,165
30 DATA 203,201,255,208,2,198,204,165,
203,197,205,208,236,165,204,197,206,20
8,230,96
40 FOR I=1536 TO 1655:READ A:POKE I,A:
NEXT I
50 DATA 104,104,133,204,104,133,203,10
4,133,206,104,133,205,162,0,169,240,32
,53,6,169,40,32,103,6
60 DATA 165,207,208,8,169,160,32,103,6
,24,144,10,169,40,32,53,6,169,120,32,1
03,6,169,240,32,53,6,96
65 DATA 133,208,161,203,41,128,168,161
,203,41,127
70 DATA 201,96,176,11,201,32,176,5,24,
105,64,208,2,233,32,129,205,152,1,205,
129,205,230,203,208,2
80 DATA 230,204,230,205,208,2,230,206,
198,208,208,209,96,133,208,169,0,129,2
05,230,205,208,2
90 DATA 230,206,198,208,208,244,96
110 P=241:POKE 207,0:POKE 82,0:OPEN #2
,4,0,"E:":T$(1)=".":T$(480)=".":T$(2)=
T$
120 SCR=PEEK(88)+256*PEEK(89)+120:LL=7
0:LM=5:IND=5:TAB=49:PS=66:FF=6:GOTO 30
00
290 ? " INSERT TEXT OR ... PRESS SELEC
T TO EDIT"
295 POKE 702,0:POKE 752,0
300 POSITION 0,0:? SIZ-LEN(T$);" FREE"
:S=STICK(0):IF S=15 THEN 330
305 IF S=14 AND P<LEN(T$)-320 THEN P=P
+40
310 IF S=13 AND P>280 THEN P=P-40
315 IF S=11 AND P<LEN(T$)-280 THEN P=P
+1
320 IF S=7 AND P>241 THEN P=P-1
330 A=USR(1536,ADR(T$)+P-241,SCR)
335 K=0
340 POKE 53279,8:PK=PEEK(53279):IF PK=
5 THEN GOSUB 900
350 IF PK=3 THEN 500
360 IF PEEK(764)<255 THEN 400
365 K=K+1:IF K<10 THEN 340
370 IF STRIG(0)=0 THEN P=LEN(T$)-240:P
OKE 207,0
380 GOTO 300
400 POSITION 0,10:INPUT #2;I$:PK=PEEK(
207):IF PK=0 THEN A$=""
405 LI=LEN(I$):LT=LEN(T$):IF LI=0 THEN
 460
407 IF LI+LT>SIZ THEN POSITION 0,1:? "
OUT OF SPACE":GOTO 300
410 IF PK=1 THEN A$=T$(P,P+39):IF T$(P
+39,P+39)=" " THEN I$(LI+1)=" ":LI=LI+
1
420 LA=LEN(A$):AD=ADR(T$):IF LI>LA THE
N A=USR(ADR(S$),AD+LT-1,AD+P-2,LI-LA)
430 T$(P,P+LI-1)=I$
440 IF LA>LI THEN T$(P+LI)=T$(P+LA)
450 P=P+LI:T$(LT+LI-LA+1)="":POKE 207,
0:GOTO 300
460 IF PEEK(207)=1 THEN 470
465 IF P<LEN(T$)-279 THEN T$(P)=T$(P+4
0)
470 POKE 764,255:GOTO 300
500 TRAP 950:ST=PEEK(560)+PEEK(561)*25
6+4:POKE ST-1,70:POKE ST+2,7:POKE ST+3
,112:POKE ST+4,6:POKE ST+5,6
505 POKE ST+24,65:POKE 712,148
510 POKE ST+25,PEEK(560):POKE ST+26,PE
EK(561)
515 OP=OP+1:IF OP=7 THEN OP=1
520 POKE 752,1:? CHR$(125):POSITION 20
,0:IF OP=1 THEN ? "LOAD "
522 IF OP=2 THEN ? "EDIT"
534 IF OP=3 THEN ? "PRINT"
536 IF OP=4 THEN ? "SAVE"
538 IF OP=5 THEN ? "DISPLAY"
539 IF OP=6 THEN ? "REPLACE"
540 POSITION 0,1:? "PRESS START TO BEG
INOPTION TO ALTER MODE"
550 FOR D=1 TO 30:NEXT D
555 POKE 53279,8:IF PEEK(53279)=3 THEN
 515
557 IF PEEK(53279)<>6 THEN 555
560 POKE 764,255:POSITION 20,1:? CHR$(
125):POSITION 0,1:ON OP GOTO 2000,290,
590,1500,590,5000
590 FOR I=1 TO 6:? ";:NEXT I:? :FOR I=
1 TO 6:? "       ";:NEXT I:? "  "
594 POSITION 0,1:? "SET FORMAT CONTROL
S":POSITION 0,6:? "LINE  LEFT    IN-
   TAB     PAGE  FORM":POSITION 0,7
595 POKE 752,0
596 ? "SIZE  MARG    DENT    STOP    S
IZE  FEED";:? "  ";LL;"";LM;"";IND;"";
TAB;"";PS;"";FF:POSITION 0,8
600 INPUT LL,LM,IND,TAB,PS,FF:P=240
605 POKE 702,64:IF OP=5 THEN 710
610 ? "DO YOU WANT TO PAUSE AT END OF
EACH PAGE    (Y/N)";:INPUT PL$:IF PL$<
>"Y" AND PL$<>"N" THEN 610
710 LINE=0:GRAPHICS 0:POKE 712,148:POS
ITION 0,3:FL=0:POKE 752,1:? CHR$(125)
715 RL=LL:TP=P:B=ASC(T$(TP,TP))
720 RL=LL-IND*(B=9)-TAB*(B=20)
725 IF B=19 AND OP=3 AND LINE<=(PS-FF)
 THEN LPRINT " ":LINE=LINE+1
726 IF B=19 AND OP=5 THEN ?
727 IF B=16 AND OP=3 THEN FOR I=1 TO P
S-LINE:LPRINT " ":NEXT I:LINE=0:GOSUB
870
728 IF B=16 AND OP=5 THEN ? :? :? :LIN
E=0
735 C=0:K=0
740 K=K+1:TP=TP+1:IF K=RL+1 THEN 765
745 IF TP>LEN(T$)-241 THEN FL=1:GOTO 8
10
750 A=ASC(T$(TP,TP)):IF A<32 THEN C=0:
GOTO 780
755 IF A=32 THEN C=C+1
760 GOTO 740
765 IF C=0 THEN A$=T$(P+1,TP-1):TP=TP-
1:GOTO 810
767 IF T$(TP,TP)=" " THEN A$=T$(P+1,TP
-1):GOTO 810
768 IF T$(TP-1,TP-1)=" " THEN C=C-1
770 K=1
775 TP=TP-1:IF T$(TP,TP)<>" " THEN K=K
+1:GOTO 775
780 IF TP=P+1 THEN P=TP:GOTO 715
785 A$="":I=P+1
790 A$(LEN(A$)+1)=T$(I,I):IF T$(I,I)<>
" " THEN 805
795 IF C>1 THEN A=INT(K/C+RND(0)):IF A
>0 THEN FOR J=1 TO A:A$(LEN(A$)+1)=" "
:NEXT J:K=K-A
800 C=C-1
802 IF C=1 AND K>0 THEN FOR J=1 TO K:A
$(LEN(A$)+1)=" ":NEXT J
805 I=I+1:IF I<TP THEN 790
810 POKE 77,0:IF FL THEN A$=T$(P+1,TP-
1)
815 IF OP=3 THEN LINE=LINE+1:IF LINE>(
PS-FF) THEN LINE=1:FOR I=1 TO FF:LPRIN
T " ":NEXT I:IF PL$="Y" THEN GOSUB 870
820 SP=LM+(B=9)*IND+(B=20)*TAB+(B=3)*(
LL-LEN(A$))/2:IF SP>40 THEN SP=40
830 IF OP=3 THEN LPRINT SP$(1,SP);A$
```

```
840 IF OP=5 THEN ? SP$(1,SP);A$
845 POKE 53279,8
850 IF FL THEN ? "▼▼          END OF P
RESENT TEXT.":? "▼        PRESS START
TO CONTINUE."
855 IF FL THEN IF PEEK(53279)<>6 THEN
855
856 IF FL THEN 500
860 P=TP:GOTO 715
870 POKE 53279,8:? "PAPER OUT : START
WHEN READY"
875 IF PEEK(53279)=6 THEN GRAPHICS 0:P
OKE 712,148:RETURN
880 GOTO 875
900 PK=PEEK(207):IF PK=1 THEN POKE 207
,0:GOTO 930
910 IF PK=0 AND P<LEN(T$)-279 THEN POK
E 207,1
930 A=USR(1536,ADR(T$)+P-241,SCR):FOR
D=1 TO 50:NEXT D:RETURN
950 ? "ERROR ";PEEK(195);" AT ";256*PE
EK(187)+PEEK(186):GOSUB 5200:GOTO 500
1500 POKE 702,64
1501 ? " ENTER FILE-NAME ";:INPUT I$:O
PEN #3,8,0,I$:N=INT(LEN(T$)/128):? #3,
N:IF N=0 THEN ST=0:GOTO 1520
1510 FOR I=1 TO N:ST=128*I:? #3;T$(ST-
127,ST):NEXT I
1520 ? #3;T$(ST+1,LEN(T$)):CLOSE #3:GO
TO 500
2000 POKE 702,64
2001 CLOSE #3:? " ENTER FILE-NAME ";:I
NPUT I$:OPEN #3,4,0,I$:INPUT #3,N:IF N
=0 THEN BEG=-127:GOTO 2020
2010 GRAPHICS 0:POKE 712,148:FOR I=1 T
O N:BEG=128*I-127:INPUT #3,A$:? A$;:T$
(BEG)=A$:NEXT I
2020 INPUT #3,A$:T$(BEG+128)=A$:CLOSE
#3:POKE 1536,104:GOTO 500
3000 LP$(1)=CHR$(0):LP$(100)=LP$(1):LP
$(2)=LP$:L$="":TRAP 3010:RESTORE 4020:
POKE 712,148:POKE 752,1:? CHR$(125)
3005 FOR I=1 TO 100:READ C$:LP$(I)=CHR
$(LEN(C$)):L$(LEN(L$)+1)=C$:NEXT I
3010 LTOP=I-1:PK=PEEK(195):IF PK<>0 TH
EN POP
3012 TRAP 950:IF PK<>0 AND PK<>6 THEN
950
3015 GOSUB 4000:GOTO 500
3020 DATA abundant,bred from,common,di
stributed,exceedingly,fairly,usually,s
carce,infrequent,one only
3025 DATA (confirmation required),loca
lly,most years,not,occasionally,pupae,
quite,rare,sometimes,at light
3030 DATA uncommon,very,widely,most re
cent record,larvae,(dubious record)
```

```
4000 RESTORE 3020:FOR I=1 TO 26:READ C
$:D$(LEN(D$)+1)=CHR$(LEN(D$)+LEN(C$)+2
):D$(LEN(D$)+1)=C$:NEXT I:RETURN
4020 DATA Warburton's Wood,Whitegate W
ay,Pettypool Wood,Winsford,Delamere Fo
rest,Winsford (A54)
4021 DATA Hogshead Wood,Little Budwort
h Common,Lea Green,Vale Royal Cut (Win
sford),Christchurch (Dorset)
4022 DATA New Forest (South Hampshire)
,Bournemouth (Dorset),Sherborne (Dorse
t),Beaumont (near Windsor)
4023 DATA Newchurch Common,Rossett (cl
wyd),Runcorn,Formby (Lancs.),Kennel Wo
ods,Darnall Woods,Congleton
4024 DATA Rookery Pool,Dutton,Dungenes
s (Kent),Crewe,Cat's Clough,Moreton (W
irral),Hawkstone (Salop.)
4025 DATA Marford (Clwyd)
4026 DATA Wharton Heath,Whixall Moss (
Salop.),Alderley Edge,Portslade (Susse
x),Streatley (Berks.)
4027 DATA Bromley (Kent),Sherrat's Rou
gh,Dutton Hollows,Arthog (Gwynedd),Spi
tal Dam (Wirral),Weaverham
4028 DATA Risley Moss,Widnes,Solihull
(Warwicks.)
5000 POKE 559,0:IF LEN(T$)=SIZ THEN GO
SUB 5210:? "OUT OF SPACE":GOTO 500
5004 XT=LEN(T$):T$(XT+1)=" ":T$(SIZ)="
 ":T$(XT+2)=T$(XT+1)
5005 J=SIZ+1:FOR I=XT TO 1 STEP -1:J=J
-1:T$(J,J)=T$(I,I):NEXT I:I=1:ERR=0
5010 A=ASC(T$(J,J)):IF A>127 THEN 5035
5015 T$(I,I)=T$(J,J):I=I+1
5020 J=J+1:IF J<=SIZ THEN 5010
5025 IF PEEK(559)=0 THEN GOSUB 5210
5026 IF ERR>0 THEN ? ERR;" ERRORS IN R
EPLACEMENT CODES":GOSUB 5200
5030 T$(I)="":? CHR$(253):GOTO 500
5035 IF (A>158 AND A<176) OR (A>185 AN
D A<193) OR (A>218 AND A<224) THEN T$(
I,I)=CHR$(A-128):I=I+1:GOTO 5020
5037 IF A>175 AND A<186 THEN 5100
5040 IF (A>192 AND A<219) OR (A>224 AN
D A<251) THEN 5050
5045 ERR=ERR+1:GOTO 5015
5050 LC=32:IF A>223 THEN A=A-LC:LC=0
5055 K=1:FOR A=A-192 TO 1 STEP -1:L=K:
K=ASC(D$(K,K)):NEXT A
5060 C$=D$(L+1,K-1):IF LEN(C$)+I>=J TH
EN GOSUB 5210:? "OUT OF SPACE":GOSUB 5
200:GOTO 5070
5062 IF ASC(C$(1,1))<97 THEN LC=0
5065 C$(1,1)=CHR$(ASC(C$(1,1))-LC):T$(
I,I+LEN(C$)-1)=C$:I=I+LEN(C$):GOTO 502
0
```

```
5070 FOR J=J TO SIZ:T$(I,I)=T$(J,J):I=
I+1:NEXT J:GOTO 5025
5100 C$="":FOR K=0 TO 2:IF ASC(T$(J+K,
J+K))<176 OR ASC(T$(J+K,J+K))>185 THEN
 POP :GOTO 5110
5105 C$(LEN(C$)+1)=CHR$(ASC(T$(J+K,J+K
))-128):NEXT K
5110 TRAP 5115:LOC=VAL(C$):IF LOC>=1 A
ND LOC<=LTOP THEN 5120
5115 TRAP 950:GOTO 5045
5120 TRAP 950:L=0:FOR D=1 TO LOC:K=L+1
:L=K+ASC(LP$(D,D))-1:NEXT D
5125 C$=L$(K,L):IF LEN(C$)+I>=J THEN G
OSUB 5210:? "OUT OF SPACE":GOSUB 5125:
GOTO 5070
5130 J=J-1+LEN(STR$(LOC)):T$(I,I+LEN(C
$)-1)=C$:I=I+LEN(C$):GOTO 5020
5200 ? "PRESS RETURN TO CONTINUE":INPU
T I$:RETURN
5210 POKE 559,32:GRAPHICS 0:POKE 712,1
48:RETURN
```

FLASH!!

Atari has sent us a new Atari 800XL to test and review! It is a very nice computer with 64K RAM, built in BASIC, and many new nifty features. Full review next issue, but would make an excellent Christmas Gift. The Atari User Support Team is also questioning user groups on what kinds of things they would like to see in the expansion box.

TINY TEXT MARK II

by BILL HARDWICK

(reprinted from Page 6, a user group 'zine in England).

I am highly impressed with the performance of TINY TEXT by Stan Ockers and Jim Carr of ACE. In itself it more than justified the price of a year's subscription for me!

After I had corrected my typing errors and got it working (for those still struggling, YES! it does work as printed), I found a few rough edges and a lack of one facility which would save me considerable time at the typewriter keyboard. In case these may be of more general interest I have developed an enhanced version of this extremely practical utility, which is listed below.

THE CHANGES MADE

A number of relatively trivial changes have been made to ease operation :

1. A couple of the screen prompts were not quite as helpful as they might have been; hopefully these are now a little clearer.

2. A few minor improvements to the screen presentation have been made - eg. the background colour has been extended into the border to give a less cramped feel to the screen when in EDIT or DISPLAY modes; the screen cursor has been suppressed where unnecessary; etc.

3. Changing paper on my printer when using single-sheet stationery was a bit of a nuisance owing to the lack of a paper-low flag. There is now an option to get an automatic pause at end of page (however reached) to change paper and restart is screen-prompted.

Which leaves us with the non-trivial extension :

4. My main recreational interest is entomology. This entails the production of records of species encountered for my own interest and that of various organizations. A characteristic of these lists is that a number of descriptive words and phrases tend to occur very frequently, and the same locality name crops up over and over again.

What I needed therefore was a simple short code for each of these items, which TINY TEXT would then "translate" to the appropriate representation automatically. This has now been provided in two forms by the REPLACE option described below.

TAILORING THE PROGRAM TO YOUR OWN REQUIREMENTS

As listed the program contains replacement codes tailored to my needs. Unless you happen to be interested in wild-life recording and resident in Cheshire this is unlikely to be of much benefit to you! Fortunately it is very easy to alter the program to give your own bespoke version which you can then save for future use.

Remember the caution and either type CLR or temporarily alter the DIM of T$ to say FRE(0)-500 and reinstate it after you have done your other changes. If you forget to do this you will probably get ERROR 2 when editing the program. A further note of caution to those of you who may already have keyed up the original program — the machine code routines at lines 40-90 have also been changed, so check these carefully; you'll get an interesting but decidedly unhelpful effect when using the replacement codes if you don't make these changes!

There are two types of replacement codes; the details on their use is given below. To change them you must follow these simple rules :

Codes starting at line 3020: there must be exactly 26 replacement strings here and the total length of all these string literals must not exceed (255-26) = 229 characters. Otherwise change them to whatever you wish.

Codes starting at line 4020: The total number of replacement strings given here must not exceed the DIMENSION of LP$. Also the total length of these strings must not exceed the DIMENSION of L$. As coded the program allows for 100 replacement codes whose total length should not exceed 1500 characters; this allows quite a bit of "slack" for expansion later without redimensioning. If you wish to vary these limits, change line 13 appropriately — the rest of the program will take care of itself. Note that there is an absolute limit of 999 codes of this type in-built into the program logic in line 5100, though this too could easily be changed by altering the FOR loop to FOR K = 0 TO 3.

As a guide, the program as written gives a text buffer of 25 Kbytes on a 48K machine without DOS.

OPERATING INSTRUCTIONS

The control characters during EDIT mode are repeated here; they all require simultaneous use of the CTRL key plus a mnemonic letter:

P = Throw to head of new page
E = End current line and start a new one
I = Start a new line indented
T = Start a new line at the TAB setting
S = Skip one blank line
C = New line with text centred in the line.

EDIT MODE ADDITIONAL INSTRUCTIONS

All the work you have to do for automatic replacement of the code values is done in EDIT mode. Both types of code are identified to the program in the same way, by typing the code in inverse video mode (ATARI key). The two types of code can be mixed freely, and there is no need to come out of inverse video simply to put spaces, punctuation or"special characters"(ie. !"#$•&'@()×†-=;:+*,./[]?) between successive replacement codes. All such characters the program encounters in their inverse video form will simply be translated to their normal form by the REPLACE option. This has been included to further reduce the number of key-strokes necessary to achieve the desired result. Note however that any of the control characters defined above must be presented in normal video form.

1. ALPHABETIC CODES. The 26 DATA items starting at line 3020 are each in turn associated with successive letters of the alphabet. This gives you an opportunity to select easily remembered mnemonics for many items, which are quickly learned and therefore soon add to the typing speed.

As a further option the code letter may be entered in either lower or upper case inverse video. If it is in upper case and the replacement value starts with a letter, then the first letter of the first word of the replacement string will also appear in upper case; this is for convenient incorporation of replacement codes at the start of a sentence.

2. NUMERIC REPLACEMENT CODES. The values starting at line 4020 are each in turn associated with a successive number starting at 1 in inverse video. All the same remarks apply to these as to alphabetic codes with two exceptions :

(a) There is no corresponding "upper case" feature;

(b) consecutive numeric replacement codes must be separated from each other by some other character. Failing this the program will assume a triplet of three inverse video numeric characters is a single replacement code and act accordingly.

THE REPLACE OPTION. The codes are not translated as entered in EDIT mode; to achieve this you leave EDIT and select REPLACE mode (by pressing the OPTION key as with all other options). A WORD OF WARNING: don't panic! Having pressed the START key unless you're into transcendental meditation this might be a good time to walk the dog, mow the lawn or decorate the spare bedroom! The screen will have turned uniformly blue as no doubt will your language as you become increasingly fearful for the fate of that text you so painstakingly typed in for the last hour. Be patient with it; there's a lot of work going on if your text is of any substantial length and it's all happening in BASIC. I can assure you it does work (I used it yesterday to create a document which took nearly an hour and a half to print after all the replacement etc. was done), and at least you're not tied to the keyboard doing repetitive typing while all this is going on — or making numerous typing errors in the process. To release you from your vigil at the blank TV, you will be given an attention-seeking "bell" as well as a screen prompt when all is ready.

Well that's it— everything else you need to know is covered by screen prompts, so more power to your typing finger!

GRAPHICS CHARACTERS

on the Epson MX80 with Graphtrax

Have you ever wanted to print out a program with inverse and graphics characters, or machine language programs which cause your printer to go crazy? Well, this program can help. It takes a disk file and, using 480 dots per line Bit-image graphics mode on an Epson MX80, prints any of the Atari's inverse and graphic characters along with normal text. The file must not be a BOOT file. BASIC and ASSEMBLY programs must be LISTed to disk. In Binary LOAD files, the 6 header and tail bytes will be printed along with the program.

After typing in the program, SAVE it and then RUN. A disk directory comes up and the program asks for a filename and tells you the buffer space, in bytes and sectors. Before entering a filename, be sure to turn on the interface and MX-80. Enter a file name and RETURN. Then the program asks if you want a modified SPACE character. This is only useful in long machine language routines. It changes the SPACE character (CHR$(32)) to an empty box on the printout. This makes it easier to count spaces.

After you enter Yes or No, the program asks if you want to see on the screen what the printer prints. If you enter No, the program turns off the screen DMA to get some extra speed. Then the program lets you enter a printout title. Next you are asked if you want to use a custom character set. If yes, enter the decimal Hi byte of the set and RETURN. Finally, the program enters the file into its buffer from 32768 to 40959 (the right cartridge slot). Don't worry, 400 owners, you still have the right cartridge slot RAM! After the load is complete, the program begins to print immediately. That's it!

As you will notice, this program is pretty slow and therefore is a perfect candidate for compiling. If anyone out there has a compiler and can successfully compile this program, I will be eternally grateful if I can get a copy of the compiled version. I will send a disk along with a SASE.

—Greg Menke
22500 Old Hundred Rd.
Barnesville, MD 20838

Power Transient Filter

(reprinted from the newsletter of the Atari Computer Club of Lawrence, KS)

This circuit is a powerline transient suppressor, a radio frequency interference (RFI) filter, and a power disconnect all in one box. It is based on a circuit in the Sept. 1983 issue of "Radio Electronics" magazine. Although the box can be built from the accompanying diagram, I recommend buying the magazine or looking it up in the library for other hints in the use and construction of the circuit.

RFI can come from many sources. It can cause glitches in memory, strange lines on the screen, or can even cause an unprotected computer (very rarely an Atari!) to reboot, wiping out the program in memory. The fix is an RFI/EMI filter sold by Radio Shack for $12.

Transients can be even more deadly. Lightning strikes or the starting of induction motors on the same power circuit can cause momentary voltages as high as 1000 volts. A nifty device called a metal oxide varistor will short out any voltages over 180 volts, causing a fuse to blow. And all of this happens in 35 nsecs (35 billionths of a second).

The third gremlin we're after is the power surge. After a power failure, the power company will try to maintain service. Often the line may surge several times before failing completely, or it may "hold" until power is restored. If an unattended disk drive were to start after a failure, having been left on for a BBS or other reason, it would most certainly eat whatever disk is inside. The remedy: a latching power relay, which disconnects the computer from the power line when the power fails.

Parts for this filter are easily available from Radio Shack for about $40.

The circuit itself varies from the one published in "Radio Electronics" in several respects: I've added an optional normally closed switch to turn the relay off. I wired the relay to break BOTH sides of the AC line (not just one), and wired the relay coil to release if the fuse blows.

Neatness in the wiring doesn't count. Sturdiness of the wiring does. Use #14 AWG solid wire to connect all wiring of the AC to the relay contacts, and to the receptacle. This can be acquired by purchasing about 2 feet of #14-2 Romex cable, and stripping off the outside insulation. The remainder of the wiring can be done with #18 or #20 stranded wire. All connections must be tied down and soldered well. The power line for any computer must be 100• free of interruptions. Leave no connections hanging in space, even if taped. Beware that small 25-watt soldering irons will cool if too many #14 wire connections are soldered too soon. Be prepared to wait a few minutes between each solder joint to let the iron heat back up, or use a bigger iron. Cold solder joints will cause more problems than they solve. And use only rosin core solder, not acid core or soldering paste.

Mount the parts on the aluminum panel of a 7-5/8 x 2-3/8 x 4-5/16 inch plastic utility box. This panel may be too thin for your taste: Feel free to make your own panel from heavier aluminum.

The START switch is indicated as a DPST momentary push button, which breaks both sides of the AC line to the relay. Other sources carry such a switch, but Radio Shack does not. Two options are available to those who do not have access to suppliers other than Radio Shack: First, RS does offer a DPDT center-off, momentary toggle switch (275-637). This will work fine if you don't mind the style of switch. The other option is to use TWO push buttons, BOTH of which must be pushed to latch the relay. This could provide just a little more security for your system. The START and STOP push buttons can be mounted in the box, or mounted elsewhere to remotely operate the system. Be aware, however, they are carrying 110VAC and the design of the wiring and switching should reflect this fact. Play it safe.

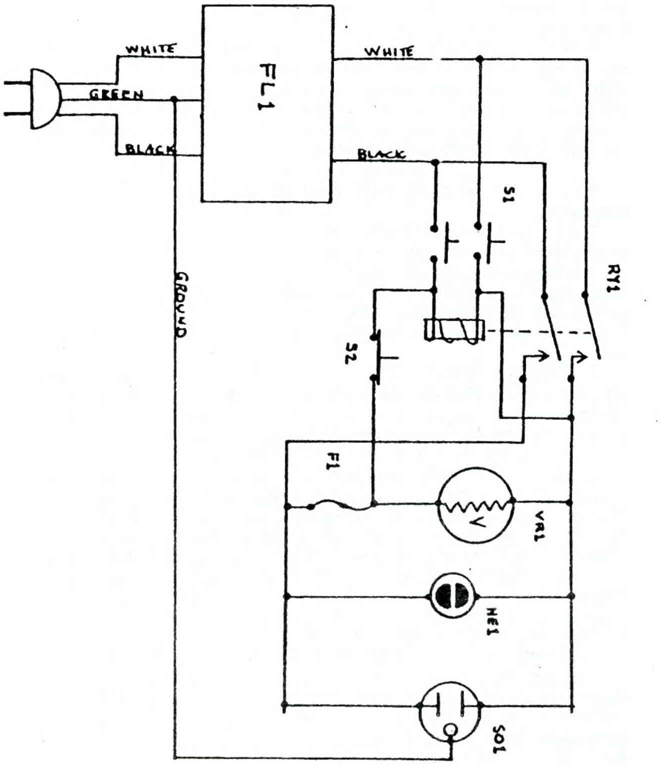
The article suggests mounting the relay socket on an aluminum bracket, mounted on the inside of the box. I mounted mine with the socket going through the box, and the relay outside. To protect your system from prying fingers, shut it off, and take the relay with you. It can't be started.

Transients which will cause the MOV to short the power will blow the fuse and disconnect the relay. Therefore, use a 10 or 12 amp 3AG fuse — but NOT a slo-blow type. Don't eliminate the fuse. It is essential.

The circuit will handle 10 amps — so count the power requirements of all portions of your computer. The original design had a single AC outlet socket: I used a standard duplex wall outlet, so the ATARI disk drive power box can be connected directly.

That should do it. Get the article and read it carefully. Then get busy and build the box. It is certainly worth the $40 and the Saturday afternoon to build it, and it can save your computer.

— Max Mayse



APE-FACE

A Printer Interface

It's no news to any of you that printer interfaces have been hard to find, and the one readily available interface requires some ROM changes some of us have been a little reluctant to make.

Well, now there are two new interfaces for you to choose from. One is a printer interface and modem combination which is made by Microperipheral Corporation. The second is the Ape-Face parallel printer interface I have been using for the last month.

The Ape-Face is small (only 4.25x3.25x1.5 inches, not including cables), so it takes up very little work space. All cables are included, and it is very easy to install. All you have to do is plug it into the I/O port of the computer or disk drive, then plug the other end into the printer and presto it's done. It is a dedicated printer interface so there are no switches to turn on when you need to use it. All you have to do is turn on the printer.

The only problem I've had in the last month is with the I/O plug. It doesn't fit very tight and it can loosen enough to break the connection. This means the computer will forget you have a printer. Or it may just print garbage.

Overall, this seems to be a very good device if all you need is a printer interface.

The Ape-Face is manufactured by Digital Devices Corp., 151 Sixth St., Suite 127 O'Keefe Bldg., Atlanta, GA 30313. — Ph. (404) 872-4430. Suggested retail $89.50.

—Bob Browning

TIDBITS

—Dale Lutz
Canada

More General Tips

Welcome to the second batch of tidbits. In this edition I'll give some ideas for speeding program entry (mostly by reducing typing).

First of all, always use abbreviations where possible. For example, I find that I can type "SE." and "SO." a lot faster than SETCOLOR and SOUND. Look in your BASIC reference manual and memorize at least some of the most common short-forms — the time it takes will be easily made up the next time you enter a program. You don't realize how good this feature is until you have to work on some machine which doesn't have it — then you more easily overlook some of ATARI BASIC's shortcomings.

If you are ever entering in a program and you come across a series of very similar lines, by all means make use of the screen editor. All you need to do is type in the first line of the series, then use the arrow keys to move up, change the line number and whatever else, and hit the return key. I even like to set tabs on the areas that need to be changed. This is a fabulous way to save time.

Did you know that all of these following lines mean the same thing to ATARI BASIC?

```
10 IF A<>0 THEN GOTO 40
10 IF A<>0 THEN 40
10 IF A THEN 40
```

You can always get away without putting GOTO after THEN if the jump is to a constant (in this case 40), but never if the jump is to a variable. The statement after THEN is always executed if the expression before it is not equal to 0. Logic operators like <,>,<>, , and = all evaluate to 1 if the expression is true, and 0 if it is false. If you think about it, you will surely see why we can shorten the IF A<>0 to simply IF A. If you're still not sure, try typing the following line in the immediate mode.

```
A=4:?A,A=4,A<>4
```

Try some more like that one if you want to.

Reading the Keyboard

This month I'll cover a few more PEEKs and POKEs and relate some methods of getting input from the computer in a program.

If you have an application where it might be nice to have the cursor disappear, you can just POKE location 752 with a 1. To turn the cursor back on, simply POKE a 0 into 752.

In many programs I have written, I have found it necessary to read the keyboard in such a way that as soon as a key is pressed, the program will take action. If a simple INPUT statement is used, the computer will wait for the return key to be pressed before it continues. There are two ways of doing this I know of. The first is a little more complicated than the second, and has both advantages and disadvantages. This method involves first OPENing an input/output control block for input from the keyboard. Then one simply uses a GET# statement to read the keyboard. This example program illustrates this technique:

```
10 OPEN #1,4,0,"K:":REM open the keyboard for input
20 GET #1,A:REM the computer will put the ASC value of the key pressed into the variable A. Note that it will not print anything on the screen.
30 PRINT CHR$(A);:REM print the character on the screen
40 GOTO 20:REM loop back for more
```

The advantage of this method is that it gives us the ASC value of the keypress, so that we don't need to do any translating. The disadvantage is that the computer will stop and wait for a keypress, which may not be any good in some situations.

The other method requires the use of the internal code of keypress location (764). First you find out the internal code of the key you want the program to react to, then you simply test the value in location 764 to see if it is the one or ones that you want. To find the internal code, use this simple immediate-mode program:

```
FOR A=0 TO 1 STEP 0:? PEEK(764):NEXT A
```

Now push the keys you want to use, and record the values the computer prints. Then in your program you just need to PEEK 764, compare it to the value or values you wanted you wanted, and branch accordingly. With this method the computer will never stop executing the program to wait for the keypress.

Reading the console keys (START, SELECT, and OPTION) is similar to the second method of reading the keyboard. In fact, one does the same thing exactly except the location used is 53279. Incidently, to clear location 764, POKE it with a 255, and to clear location 53279, POKE it with an 8. To make the little speaker inside the computer click, POKE 53279 with 7.

Next time I promise to deal with disk drive tricks.

—Dale Lutz
Canada

Meetings

Now at the South Eugene High School Cafetaria

Weds Dec 14
Features Mr Television & How to Maintain your Atari!
Also Demos of the New
Atari 800XL,
Movie Maker and More!!

Next Meeting Jan 11
Same Place- No Newsletter
Both at 7:30 PM

Best of ACE books

Volume 1 are bound issues of the ACE Newsletter from the first issue, Oct 80 to June of 1982

Volume 2 covers July 1982 to June 1983

Only $12 each ($2 extra for Airmail). Available only from:

George Suetsugu
45-602 Apuapu St
Kanoehe, HI 96744

SortFinder 1.2
Composite index of Atari related articles from 5 popular computer periodicals from Apr '81 to June '83, including ACE. Only $6 for ACE member from:

Jim Carr, Valley Soft
2660 S.W. DeArmond
Corvallis, OR 97333

Atari
Computer
Enthusiasts
3662 Vine Maple Dr. Eugene OR 97405

FIRST CLASS MAIL

Atari Computer Enthusiasts

A.C.E. is an independent, non-profit and tax exempt computer club and user's group with no connection to the Atari Company, a division of Warner Communication Company. We are a group interested in educating our members in the use of the Atari Computer and in giving the latest News, Reviews and Rumors.

All our articles, reviews and programs come from you, our members.

Our membership is world-wide; membership fees include the A.C.E. Newsletter. Dues are $10 a year for U.S., and $20 a year Overseas Airmail and include about 10 issues a year of the ACE Newsletter.
Subscription Dep't: 3662 Vine Maple Dr., Eugene, OR 97405.

**President— Kirt Stockwell, 4325 Sean , Eugene, OR 97402 503-689-5355
Vice Pres— Larry Gold, 1927 McLean Blvd., Eugene, Or 97405 / 503-686-1490
Secretary— Charles Andrews, POB 1613, Eugene, Or 974401613 503-747-9892
Librarian— Ron and Aaron Ness, 374 Blackfoot, Eugene, Or 97404 (503)689-7106.
Editors— Mike Dunn, 3662 Vine Maple Dr., Eugene, Or 97405 503-344-6193
Jim Bumpas, 4405 Dillard Rd., Eugene, Or 97405 503-484-9925
E.R.A.C.E. (Education SIG Editor)— Ali Erickson, 295 Foxtail Dr., Eugene, Or 97405 /503-687-1133
E.R.A.C.E. Corresponding Secretary— Robert Browning, 90 W. Myoak, Eugene, OR 97404 (503)689-1513.

Send 27 cents in stamps or coin (50c overseas) to the Ness' for the new, updated ACE Library List —new in Oct!!

Bulletin Board (503) 343-4352

On line 24 hours a day, except for servicing and updating. Consists of a Tara equipped 48K Atari 400 with a TARA keyboard, 2 double-density double sided disk drives with an ATR 8000 interface, 2 double density Percom disk drives, an Epson MX80 printer, a Hayes SmartModem; running the ARMUDIC Bulletin Board software written by Frank L. Huband, 1206 N. Stafford St., Arlington, VA 22201. See the Nov '82 issue for complete details.